
Multi-Clustering

An Application Note

www.arium.com

Overview

American Arium currently supports multi-clustering for Intel 32-bit processors, including the Intel® Pentium® II/III, Pentium II/III Xeon™, Mobile Pentium II/III, Pentium Pro, Celeron™, and some other soon-to-be-released processors and the Intel Itanium™ processor, Intel's flagship 64-bit processor. Using our COMET ECM-20xx or ECM-30xx, you can connect to as many as four clusters in a target at one time.

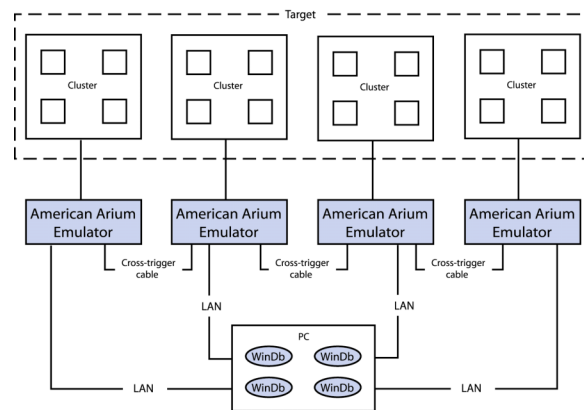
Please note that one emulator is required for each cluster. The cable(s) for multi-cluster debug support must be ordered separately. Such cables are not part of the regular emulator package.

Hardware Setup

The current methodology for supporting multi-cluster consists of connecting an ECM to each cluster's debug port via the appropriate PBD, and interconnecting these ECMs by way of "cross-trigger" cables. The cross-trigger cable, approximately three feet long, is used to connect two ECMs to one another.

To connect emulators to a multi-cluster target:

1. Connect a cross-trigger cable to the mini-DIN socket labeled "TO NEXT" on one emulator and to the mini-DIN socket labeled "FROM PREV" on the other.
2. To connect a third and/or fourth emulator, follow the instructions above.



Sample connection

Caution: Do not connect the emulators into a ring.

To connect the emulator to the target and a host (or a network supporting a host), see the "Getting Started" installation guide that came with the emulator.

Software Setup

WinDb support for multi-cluster requires that a separate copy of WinDb be running for each emulator. The separate WinDb programs communicate with each other, however, so that when a **Go** command is issued within one WinDb, the other WinDb programs automatically issue their own **Go** commands.

Option #1 – Multiple installs of WinDb

Install WinDb multiple times, once for each cluster.

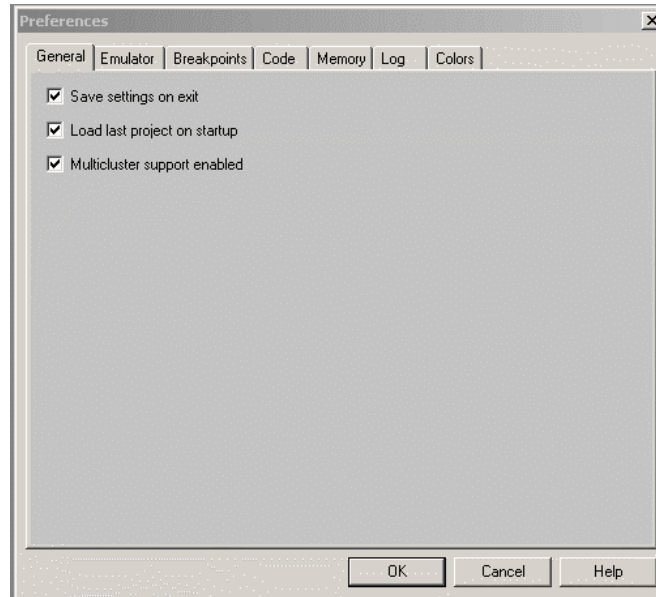
Option #2 – Single install of WinDb

1. Install a single copy of WinDb.
2. Create an “ini” file for each copy of WinDb. The standard “ini” file is “wdb.ini” (located in the directory where WinDb is installed). Name the new ini files “wdb2.ini”, “wdb3.ini”, etc. (Do not use quotation marks around the file names.)
3. Create separate icons for each copy of WinDb.
4. Associate each WinDb icon with the correct “ini” file. For each icon, right click, then select **Properties**, then **Shortcut**, and add “-ini iniFileName” at the end of the target field (e.g., -“ini wdb2.ini”).
5. Start WinDb for the first cluster. Specify the connection for the emulator controlling the first cluster. (Select **Options|Emulator Connection**.)
6. Select **File|Project|Save As** to create a project file with the current emulator connection. Pick different project file names for each cluster.
7. Repeat steps 2 and 3 for the other emulators. The goal is to create a project file for each copy of WinDb that will be running. Each project file will have a different emulator connection depending on which emulator is being controlled.

Running in multi-cluster mode

To run in multi-cluster mode:

1. Double-click on each WinDb icon to start as many copies of WinDb as there are emulators.
WinDb checks for other active WinDb tasks, and automatically registers with them. (Wait until the first copy of WinDb is up and running before double-clicking on the icon for the second copy of WinDb. This assures that each copy of WinDb will recognize each other.)
2. To enable multi-cluster support, go to **Options|General** on any WinDb menu.
The **Preferences** dialog box appears.
3. Go to the **General** tab.
4. Enable **Multicluster support enabled**.



Enable the multi-cluster option to initiate multi-cluster functionality.

5. Set up breakpoints in each WinDb and press Go in any WinDb.

The cross-trigger bus (as established by the cross-trigger cables) holds the combined system stopped until the last cluster is started.

Please note that when **Go** or **Stop** commands are issued in one WinDb (either from the menu, toolbar, or command line), that copy of WinDb sends messages to all the other WinDbs to generate the same command. There is no need to select **Go** or **Stop** in each copy of WinDb. Likewise, when one WinDb hits a breakpoint and triggers, all processors in all clusters stop (see triggering latency below).

Timing

The emulators are synchronized to one another via the cross-trigger bus, but some finite differential in time exists between the “start” signal arriving at the various clusters being controlled. This time differential is caused by circuit delays and by the method required for starting the cluster type. The Intel Pentium 4, Itanium, and some other Intel soon-to-be-released processor systems require JTAG operations to start the processors, while Intel P6-class processor systems require only the negation of a PREQ signal. The time differential (or slip) between starting individual clusters in a multi-cluster system is thus dependent upon the type of system being controlled.

- Maximum slip between starting multiple Intel Itanium or some other soon-to-be-released processor clusters is 65 ns plus $\frac{1}{2}$ JTAG Clock period. The typical slip for starting these clusters is 25 ns plus $\frac{1}{4}$ JTAG Clock period.
- Maximum slip between starting multiple Intel P6-class processor clusters is 50 ns; typical slip is 10 ns.
- All multi-clusters use a similar mechanism for stopping the processors, and thus only one set of stop “slip” times needs to be specified. Maximum slip between stopping multiple clusters is 50 ns; typical slip is 10 ns.
- In addition to the signals monitored by the emulators from their connected clusters, the external “BNC-IN” connector can be used to signal a stop to the multiple emulators connected by the cross-trigger cables. When signaling a stop to the multi-cluster environment, the maximum delay from “BNC-IN” to assertion of the stop on the cross-trigger bus is 40 ns; typical slip is 21 ns.
- When using the “BNC-OUT” connector to control an Agilent Technologies logic analyzer, the maximum

delay from a stop signaled on the cross-trigger bus to the “BNC-OUT” connector is 38 ns; typical slip is 23 ns.

- So long as the emulator is configured via WinDb to “listen” to the “BNC-IN” connector, the “BNC-OUT” connector will reflect an assertion at this “BNC-IN” connector. The maximum time from “BNC-IN” assertion to “BNC-OUT” assertion is 78 ns; typical slip is 44 ns.



14811 Myford Road
Tustin, CA 92780
Voice: 714-731-1661
Fax: 714-731-6344
Web: www.arium.com
E-mail: info@arium.com